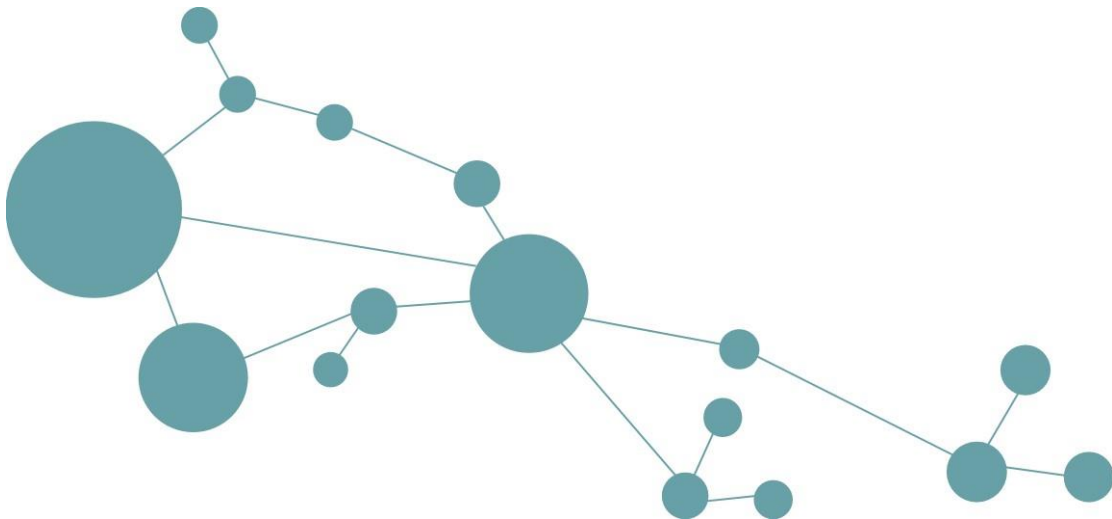
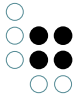




Nextbot 3.2



Inhaltsverzeichnis

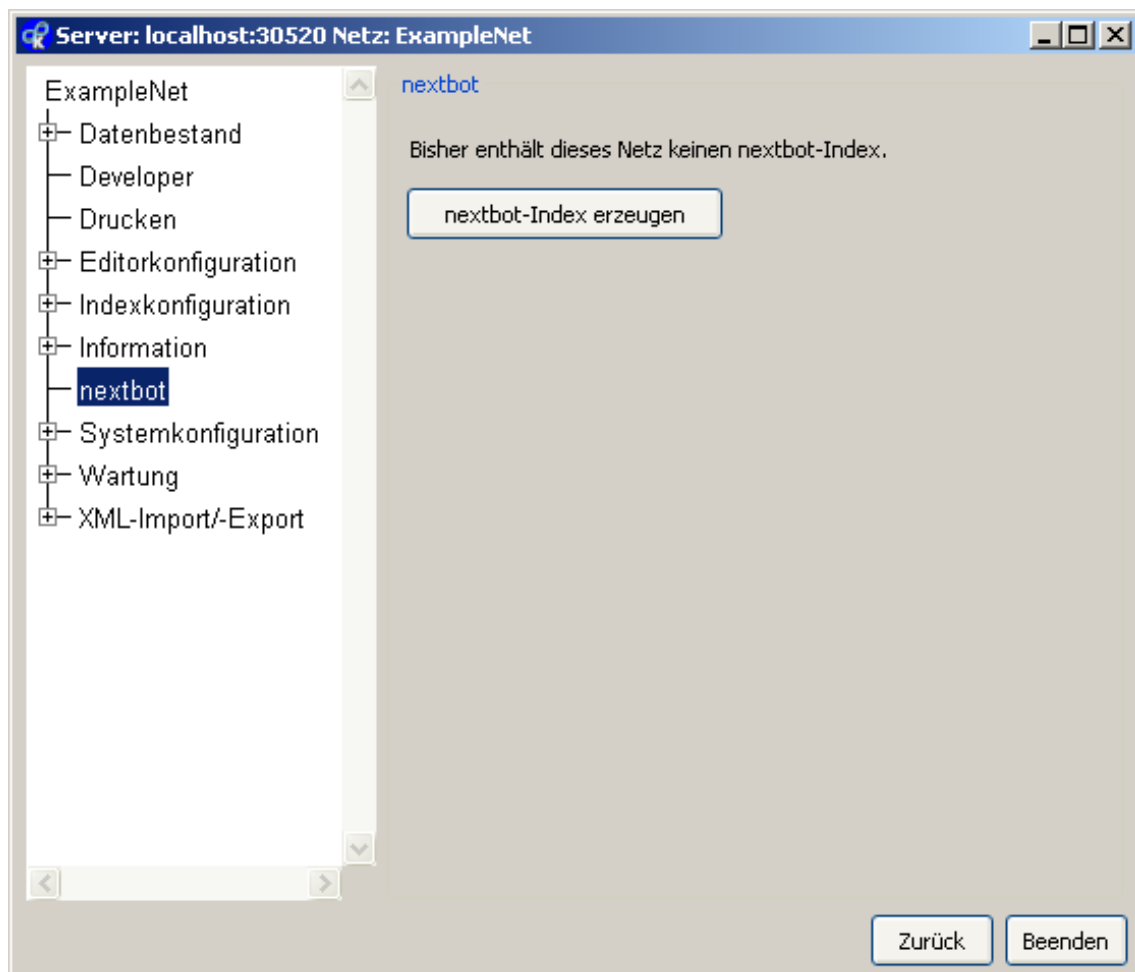
1	Administration des Nextbot-Anschlusses	3
1.1	Erzeugen des Nextbot-Index	3
1.2	URL-Ersetzung konfigurieren	12
2	Nextbot-Suchen	16
2.1	Konfiguration der Nextbot-Suche	16
2.2	Durchführen der Nextbot-Suche	17
3	KPath Funktionen	17
4	KEM	21
4.1	KEM Funktionen	21
4.2	Eigenschaften von Suchen	23
4.3	Eigenschaften von Suchergebnissen	24

1 Administration des Nextbot-Anschlusses

Dieses Kapitel erläutert alle Einstellungsmöglichkeiten des Nextbot-Anschlusses, welche in der Admin-App von K-Infinity getroffen werden können.

1.1 Erzeugen des Nextbot-Index

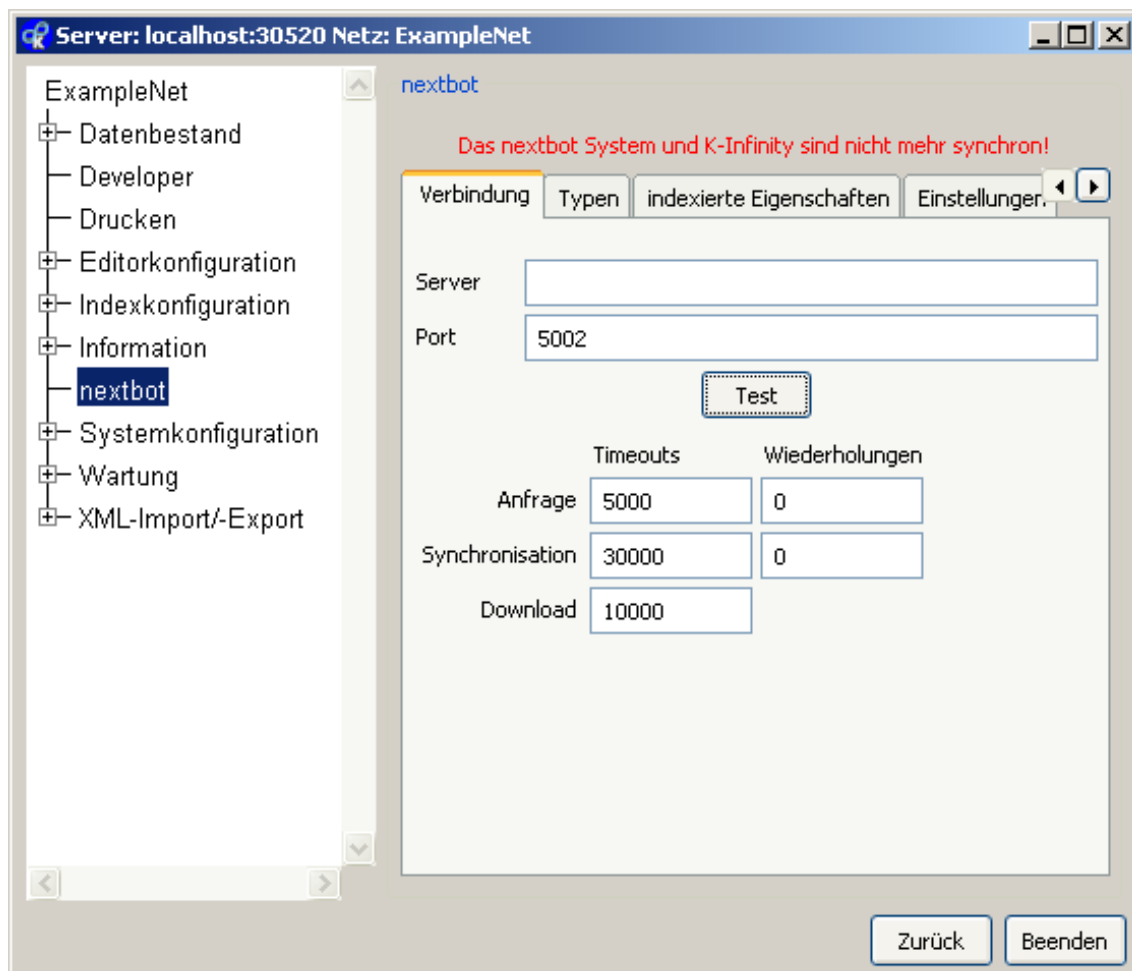
In einem neu erzeugten Wissensnetz ist anfangs zwar die Möglichkeit zur Benutzung des Nextbot-Index gegeben, dieser ist aber noch nicht konfiguriert und muss erst explizit angelegt werden. Dies geschieht durch Anwählen des Reiters 'nextbot' in der Admin-App, welcher sich zunächst wie folgt präsentiert:



Durch Klick auf den Button zur Erzeugung des Index und kurze Ab- und erneute Anwahl des Punktes 'nextbot' wird der Index angelegt und die Einstellmaske angezeigt. Da die Einstellungsmöglichkeiten des Nextbot-Anschlusses recht vielfältig sind, erscheinen sie innerhalb des 'nextbot'-Reiters auf mehrere Unterreiter gegliedert:

- Verbindung
- Typen

- indexierte Eigenschaften
- Einstellungen
- Software
- VFS
- Zugriffsrechte
- Synchronisation
- Fehlerprotokoll

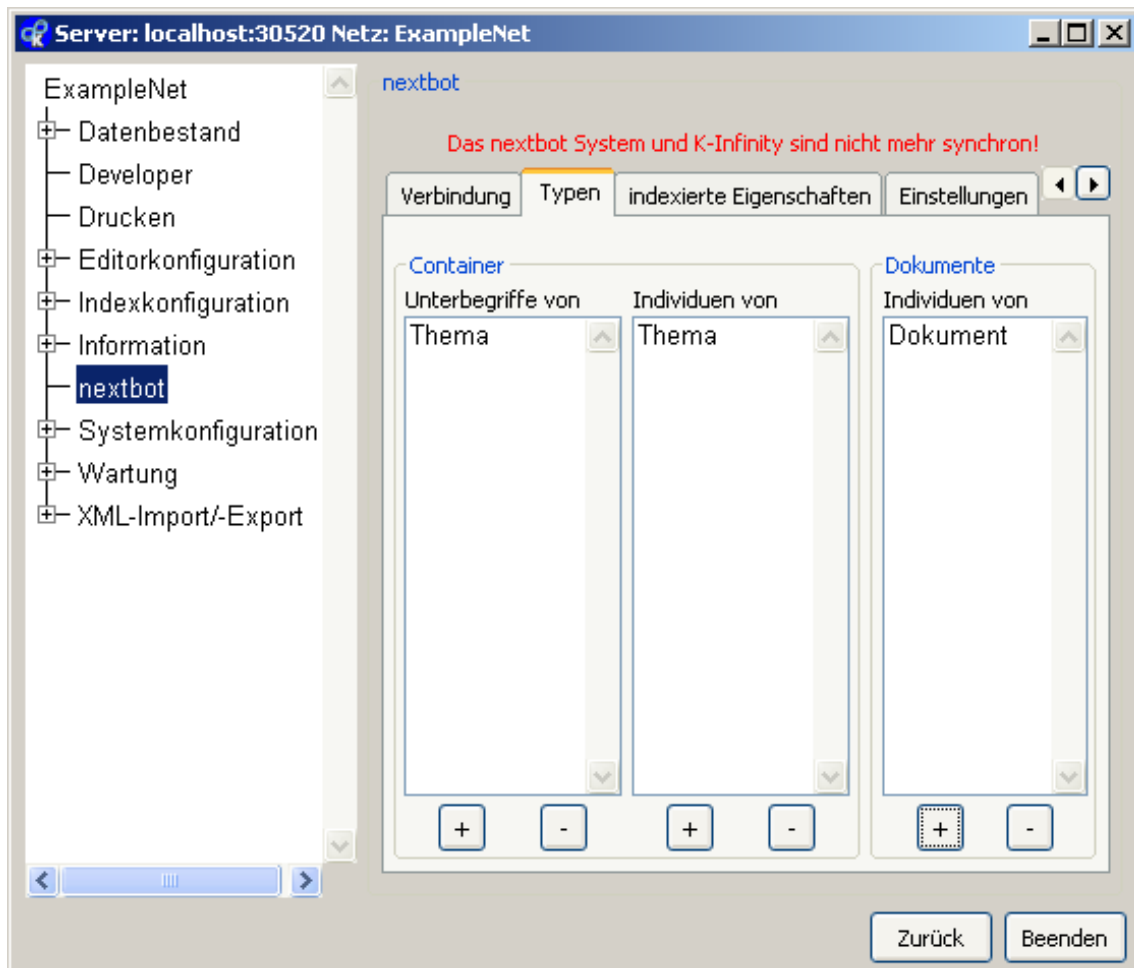


In obiger Abbildung werden die weiteren Reiter durch die Knöpfe zum seitwärts rollen sichtbar.

Auf dem dargestellten Unterreiter 'Verbindung' konfiguriert man, welcher Nextbot-Server kontaktiert wird und welche Timeouts und Wiederholversuche für verschiedene Anfragetypen gelten. Ob eine Verbindung zum angegebenen Nextbot-Server nach der Eingabe möglich ist, läßt sich durch Klicken auf den Button 'Test' überprüfen. Zu beachten ist, dass das Feld "Server" leer bleiben kann und dann alle Clients davon ausgehen, dass der nextbot Dienst auf dem gleichen Rechner läuft, wie der Mediator. Der angegeben Port muss dann auch für alle Clients, die nextbot Funktionalität verwenden erreichbar sein.

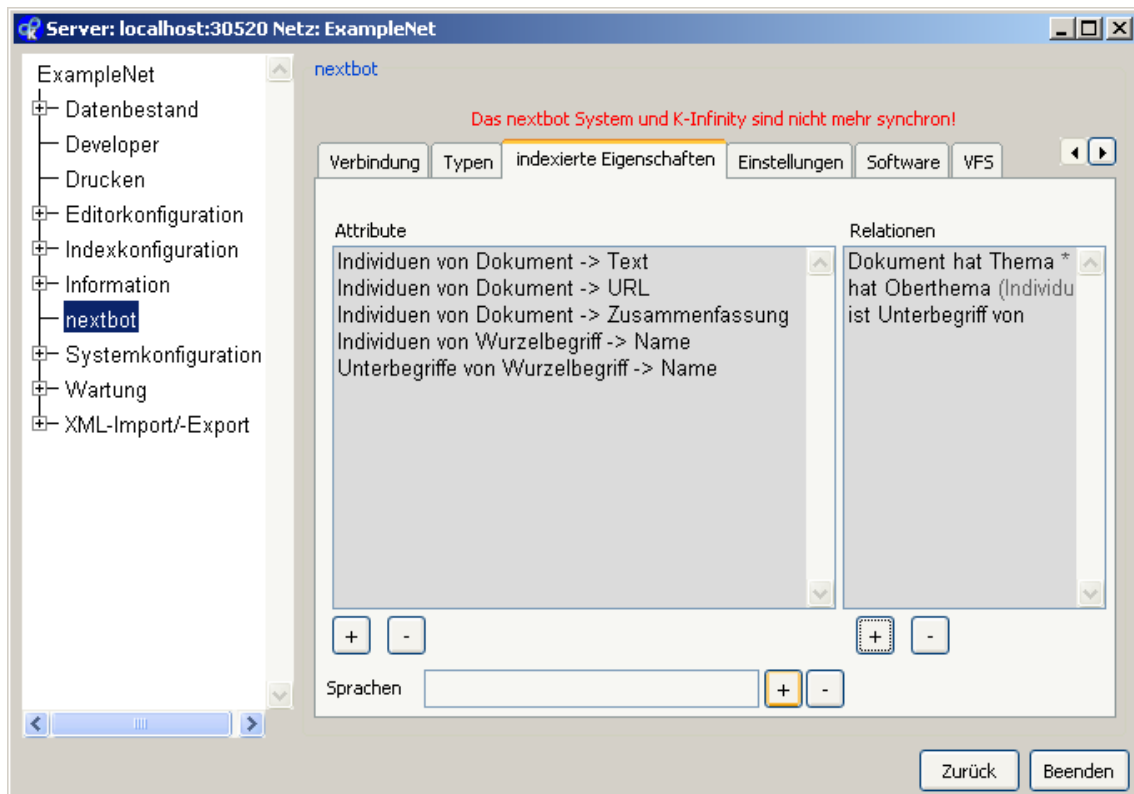
Auf dem Unterreiter 'Typen' lassen sich die Abbildungseigenschaften der Wissensnetzobjekte auf Nextbot-Objekte festlegen. Nextbot kennt die Arten Container und Dokument mit ein-

er Enthaltenseins-Relation, was man sich analog zu einem Dateisystem vorstellen kann in dem Ordner andere Ordner und Dokument enthalten. Hier wird konfiguriert, welche Wissensnetzobjekte auf welche Weise vom Nextbot-Anschluss interpretiert werden. Nur solche Wissensnetzobjekte erhalten dann auch in Nextbot eine Abbildung.



Durch Klicken auf den '+' respektive den '-' Button unterhalb des jeweiligen Abschnitts lassen sich die entsprechenden Wissensnetzbeispiele hinzufügen, bzw. entfernen.

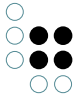
Der nächste Unterreiter 'indexierte Eigenschaften' dient dem Auswählen von Attributen und Relationen, welche mit Nextbot synchronisiert werden. Außerdem kann festgelegt werden, welche Sprachen von übersetzten Attributen indexiert werden, falls Übersetzungen verwendet werden.



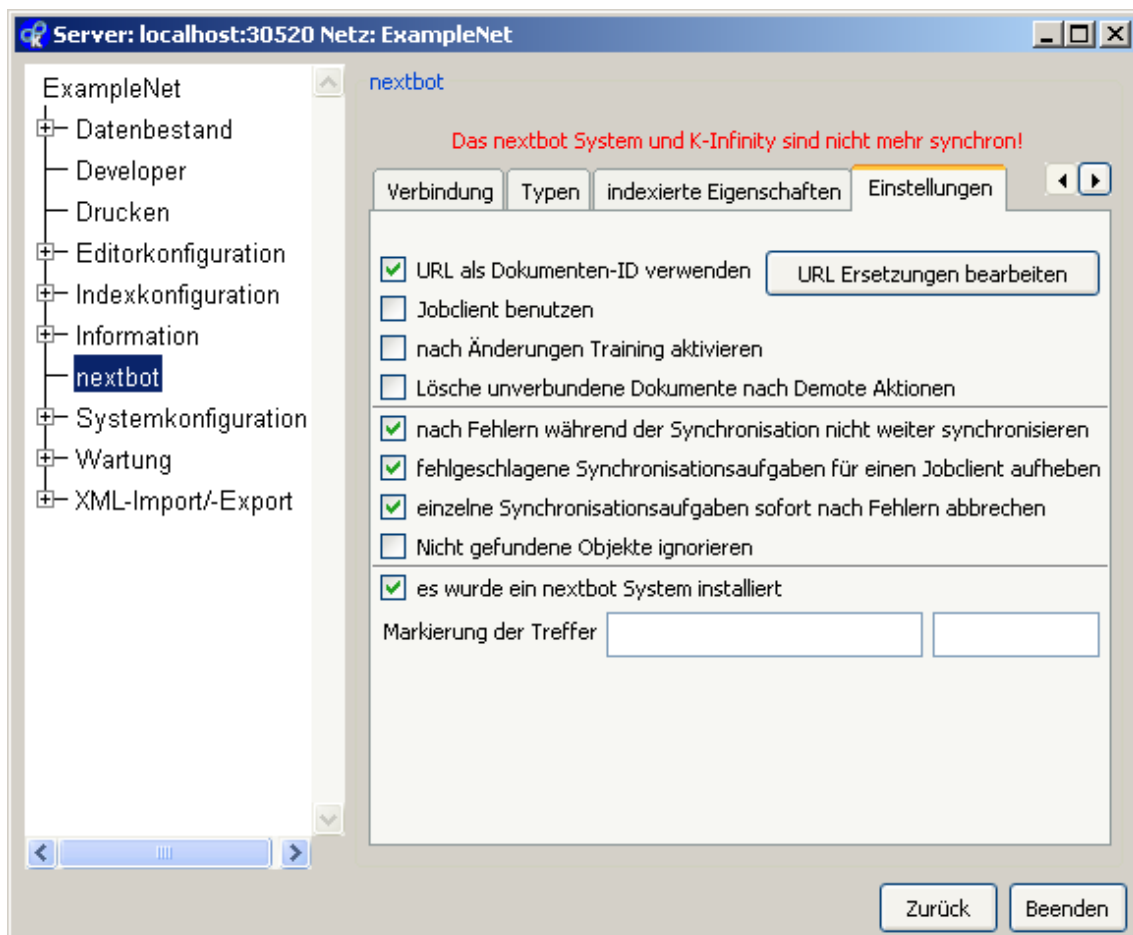
Zu beachten ist, dass mindestens ein Attribut für jedes Objekt der auf dem vorigen Reiter "Typen" erwähnten Begriffe und Individuen zu indexieren ist. Die aufgelisteten Relationen sollten die Beziehung "ist enthalten in" darstellen, also in einer Hierarchie von unten nach oben angeordnet sind; in diesem Beispiel sind es Relationen für Dokumente, die Themen zugeordnet sind und Themen, die eine Hierarchie bilden.

Im nächsten Unterreiter 'Einstellungen' werden diverse Betriebsparameter eingestellt.

- URL als Dokumenten-ID verwenden:
Muss ausgewählt werden, wenn es Dokumente geben wird, die nicht im Netz selbst vorhanden sind, sondern z.B. durch Dateisystem-Crawler gefunden werden sollen. Dies setzt natürlich voraus, dass alle Dokumente des Netzes auch ein URL Attribut haben können.
Der nebenstehende Knopf zur URL Ersetzung sollte nur in problematischen Fällen eingesetzt werden.
- Jobclient benutzen:
Diese Einstellung steuert, ob Änderungen im Netz von den Clients selbst an nextbot kommuniziert werden sollen oder ein zusätzlicher Jobclient Dienst diese Aufgabe übernimmt. Ein solcher Client muss dann so konfiguriert werden, dass er "KBrainbotJob"s bearbeitet.
- nach Änderungen Training aktivieren:
Mit dieser Option wird nach jeder Änderung innerhalb des nextbot Systems eine Reklassifikation angestoßen. Dies führt ggfs. zu einer schnelleren Anpassungen von Suchergebnissen des Ähnlichkeitssuche, kann aber Performanceverluste mit sich bringen.
- Lösche unverbundene Dokumente nach Demote Aktionen:
Wenn Dokumente als Trainingsdokumente ins Netz aufgenommen wurden, kann man durch diese Einstellung veranlassen, dass nach dem Entfernen der letzten Relation von einem Dokument zu Containern, dieses Dokument aus dem Netz gelöscht wird.



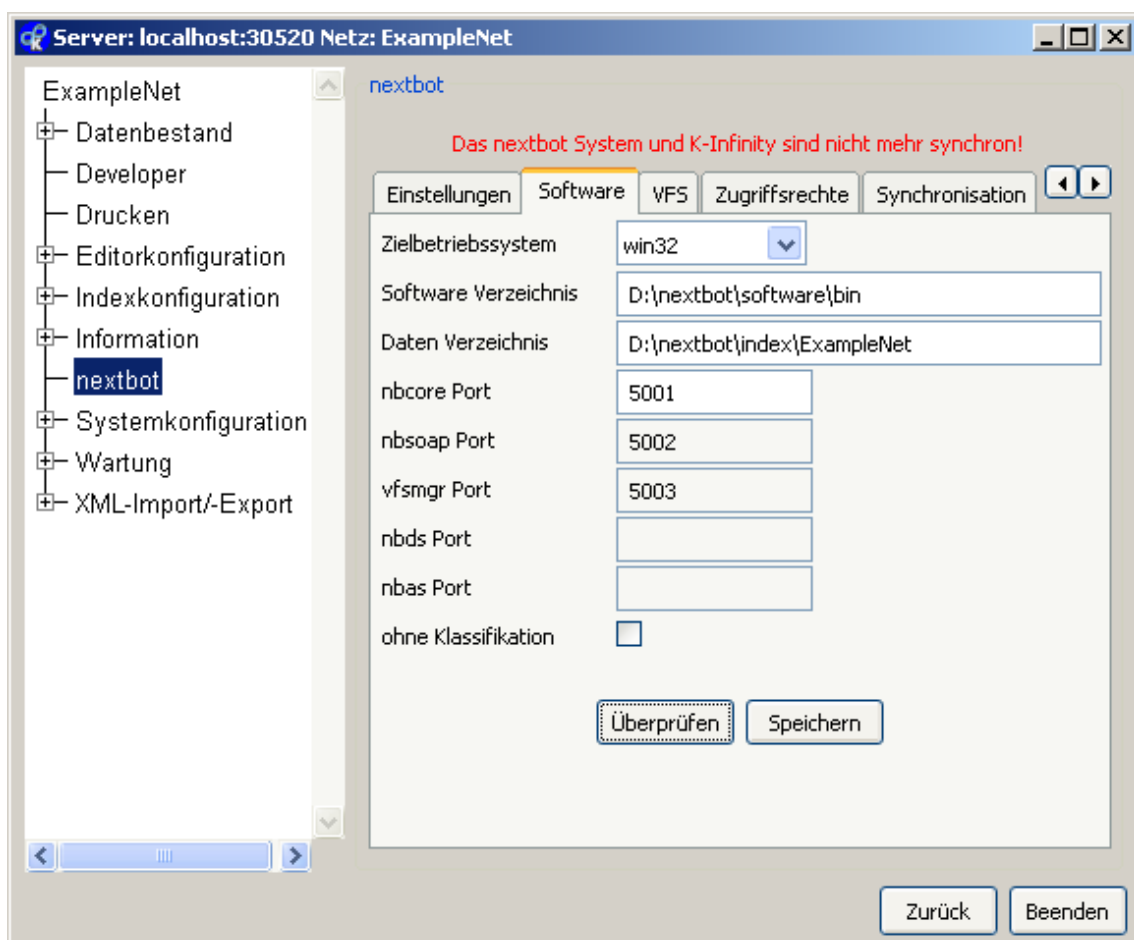
- nach Fehlern während der Synchronisation nicht weiter synchronisieren:
Wenn bei einer Synchronisation ein Fehler auftritt, wird diese und später folgende Aufgaben nicht weiter ausgeführt.
- fehlgeschlagene Synchronisationsaufgaben für einen Jobclient aufheben:
Hierdurch können Synchronisationvorgänge aufgehoben werden und ggfs. später nochmals von einem Jobclient ausgeführt werden.
- einzelne Synchronisationsaufgaben sofort nach Fehlern abbrechen:
Wenn innerhalb eines Synchronisationsvorgangs ein Fehler passiert, wird der Vorgang nicht weiter ausgeführt.
- Nicht gefundene Objekte ignorieren:
Während des Betriebs kann es dazu kommen, dass z.B. durch Löschungen in externen Systemen im Netz noch Referenzen auf solche Objekte vorhanden sind. Mit Hilfe dieser Option werden aus einer solchen Inkonsistenz resultierende Fehler ignoriert.
- es wurde ein nexbot System installiert:
Diese Option dient zum Betrieb eines KnowledgePortals ohne real ein nextbot System installiert zu haben. Dabei werden natürlich keine Ergebnisse geliefert werden können.
- Markierung der Treffer:
Diese beiden Felder dienen dazu, Trefferstellen von Sucheingaben durch nextbot mit Hilfe von eingefügten Strings markieren zu lassen und ist damit für die Konfiguration des KnowledgePortals relevant.



Der Reiter "Software" dient zum Schreiben der Konfigurationsdateien der nextbot Software. Das Zielbetriebssystem bezieht sich auf den Rechner, auf dem nextbot installiert wird. Das Software Verzeichnis ist der Ort, der dann z.B. die Datei nbversion(.exe) enthält. Das Datenverzeichnis sollte vom Software Verzeichnis getrennt sein und enthält den Volltextindex für die indexierten Daten und die Log-Dateien der laufenden nextbot Prozesse. Hier wird auch der Port der nbCore-Komponente konfiguriert, üblicherweise einer unterhalb des auf dem ersten Reiter konfigurierten nbSoap Ports. Die anderen Ports sind hier auch nur der Übersicht halber aufgeführt und werden auf anderen Reitern konfiguriert.

Der Schalter "ohne Klassifikation" schaltet effektiv die Ähnlichkeitssuche durch nextbot aus und ist in Szenarien sinnvoll, wo diese auch nicht genutzt werden soll.

Die beiden Knöpfen überprüfen einerseits die aktuelle Konfiguration (soweit das möglich ist) und speichern Konfigurations- und Startdateien.

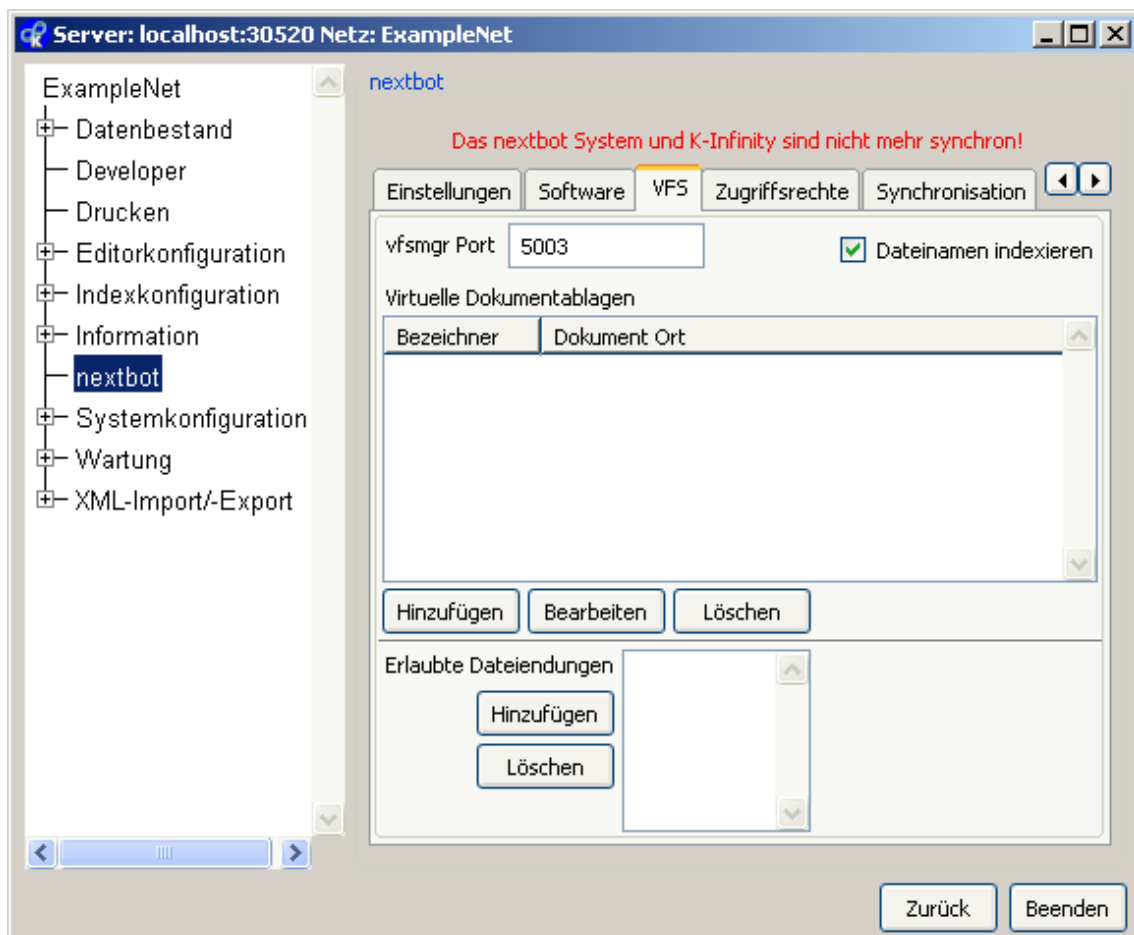


Der Reiter VFS ist relevant, wenn nextbot Verzeichnisse indexieren soll, die damit Dokumenttreffer ausserhalb des Netzes liefern werden. Wenn man dieses Feature nicht benötigt, reicht es aus, den Port auf den Wert "0" zu setzen.

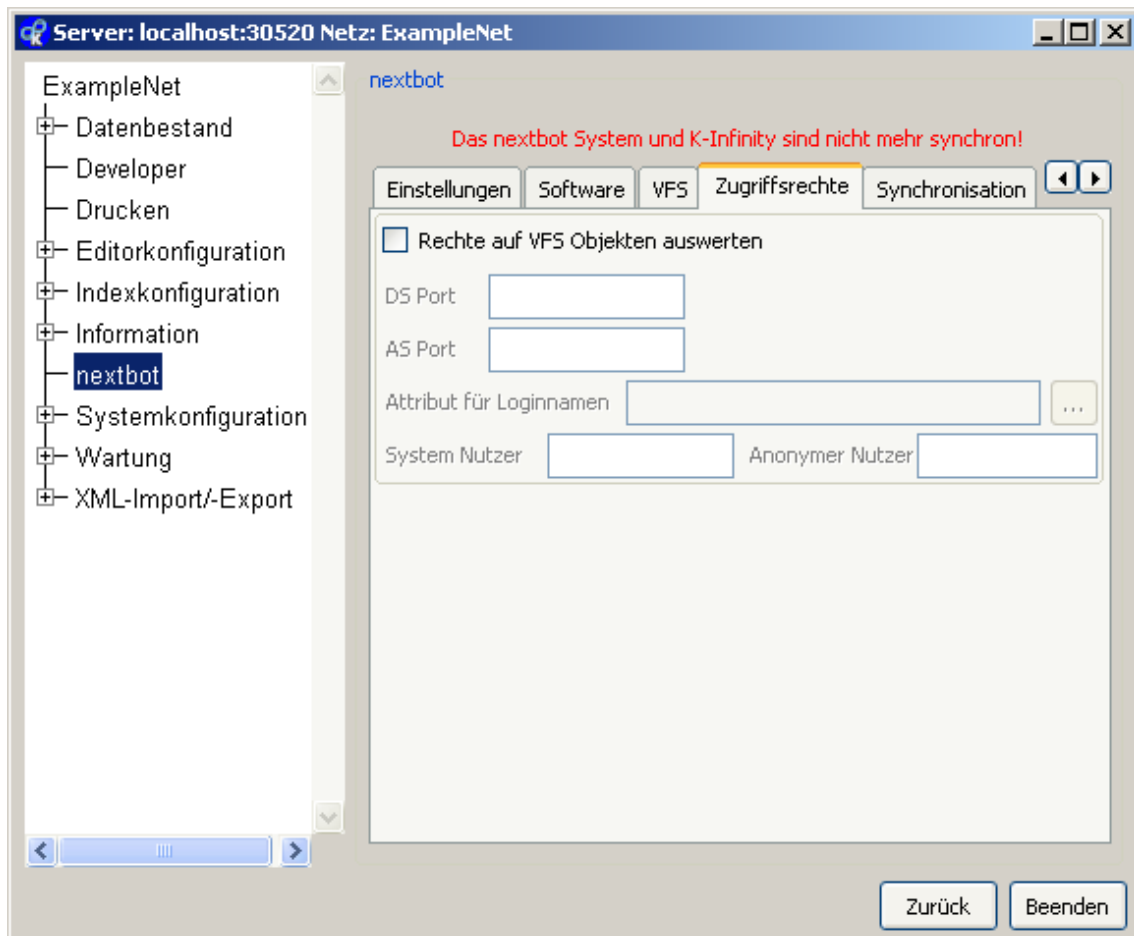
Die Option "Dateinamen indexieren" dient dazu Dateinamen durchsuchbar zu machen. Ohne diesen Schalter werden nur Dokumentinhalte indexiert.

Die Liste unterhalb dient zum Verwalten der indexierten Dokumentablagen. Hierzu dienen die direkt darunter liegenden Knöpfe. Als mögliche Typen stehen im Moment Verzeichnisse, EMail oder Web-Respositories angegeben werden, wobei der VFS selbst nur Verzeichnisse indexiert. Jedes Repository erhält einen eindeutigen Bezeichner.

Unter der Liste der Dokumentablagen findet sich eine editierbare Liste von gültigen Dateiendungen. Per Default werden alle Dateien indexiert (sofern Extraktoren dafür vorhanden sind), bei Bedarf kann über diese Liste die Menge eingeschränkt werden.



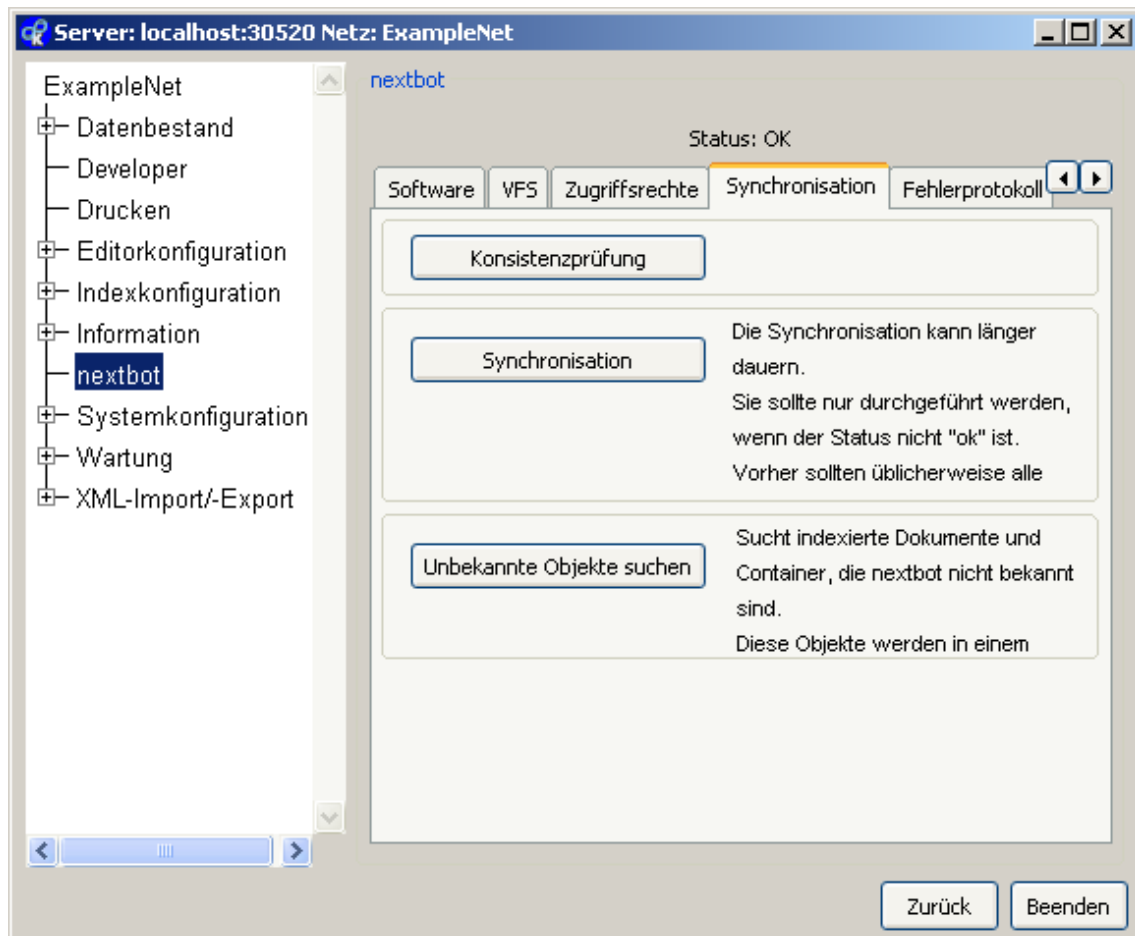
Der Reiter Zugriffsrechte ist relevant, wenn in den Dokumentverzeichnissen nicht alle Dokumente von allen Benutzern gelesen werden dürfen. Wenn man dieses aktiviert, müssen Ports für den Directory Service nbDS und Authentication Service nbAS definiert werden. An den Benutzerobjekten im Wissensnetz muss ein Attribut definiert sein, dass den Loginnamen der Nutzer in den Verzeichnisrepositories angibt. Systemnutzer und Anyonymer Nutzer werden mit Default Werten befüllt und müssen normalerweise nicht verändert werden.



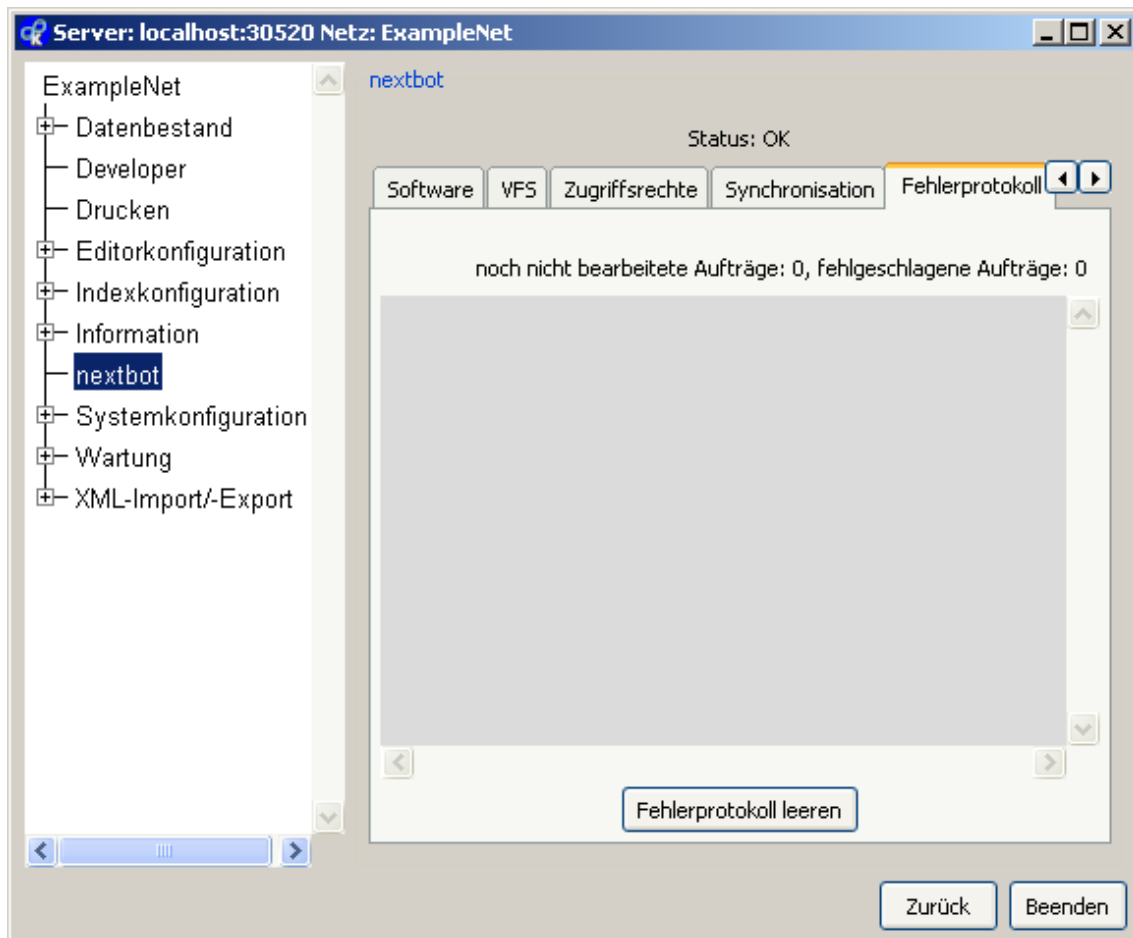
Im Unterreiter "Synchronisation" läßt sich feststellen, ob die Konfiguration Probleme aufweist ("Konsistenzprüfung"). Hierbei gemeldete Probleme sollten vor dem Betrieb bereinigt werden.

Mit dem Knopf "Synchronisation" werden nextbot Index und Wissensnetz in einen konsistenten Zustand überführt. Bevor man diesen betätigt sollte üblicherweise nextbot gestoppt werden das "index" Unterverzeichnis im Daten Verzeichnis gelöscht werden und danach nextbot neu gestartet werden. Die Dauer der Synchronisation hängt vom Datenvolumen des Wissensnetzes ab.

Nach der Synchronisation ändert sich die Kopfzeile in "OK", wie im unteren Screenshot. Sollte beim Umkonfigurieren oder im Betrieb dieser Status etwas anderes als "OK" anzeigen, muss eine Neusynchronisation durchgeführt werden.



Im Unterreiter 'Fehlerprotokoll' werden Fehlermeldungen, welche in der Kommunikation mit Nextbot auftraten, angezeigt. Ebenso werden die Anzahl ausstehender bzw. fehlgeschlagener Aufträge der Jobclient-Warteschlange angezeigt.



1.2 URL-Ersetzung konfigurieren

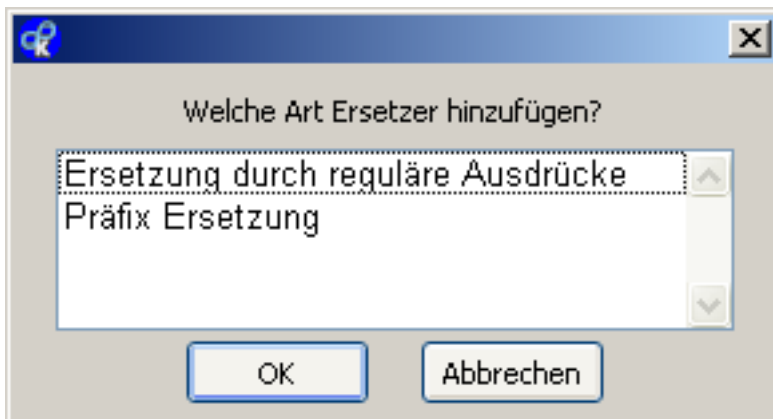
Durch Anklicken des Knopfes 'URL Ersetzungen bearbeiten' im Unterreiter 'Einstellungen' der nextbot-Konfiguration öffnet sich ein Editor, in dem sich die Transformationsregeln für URLs formulieren lassen, die für die korrekte Anbindung von externen Dokumenten via nextbot nötig sein kann.

Ein Beispiel:

Im Wissensnetz sind Dokumentrepräsentanten angelegt, welche ein URL-Attribut haben, über das sie mithilfe des Expertensuch-Index von K-Infinity identifiziert werden können. Der nextbot-Index ist angewiesen, URL-Attribute zur Identifikation von Dokumenten zu benutzen, d.h., wenn sich der Dokumentrepräsentant im Wissensnetz in den Nextbot-Index einträgt, dann überträgt er als Identifikator den Wert seines URL-Attributes. Nun kann es sein, dass dieser Wert ein anderer ist, als jener, unter welchem der Nextbot-Index die extern gehaltene Datei finden kann, die das tatsächliche Dokument darstellt, auf welches das Dokument im Wissensnetz nur verweist. Diese Notwendigkeit ergibt sich durch verschiedene Szenarien, wie der nextbot-Index an Inhalte geknüpft ist. Wenn die im Wissensnetz gespeicherte URL eine tatsächliche URL ist, dann muss keine Transformation konfiguriert werden. Ansonsten ergeben sich die URLs eventuell durch ein Schema, welches durch eine Web-Applikation gegeben ist, nextbot hat aber einen anderen, u.U. direkteren Zugang zu den Inhalten. Dann muß die allgemeine URL des Wissensnetzes auf einen Identifikator abgebildet werden, mit dem der nextbot-Index in der Lage ist, die externe Datei zu finden.

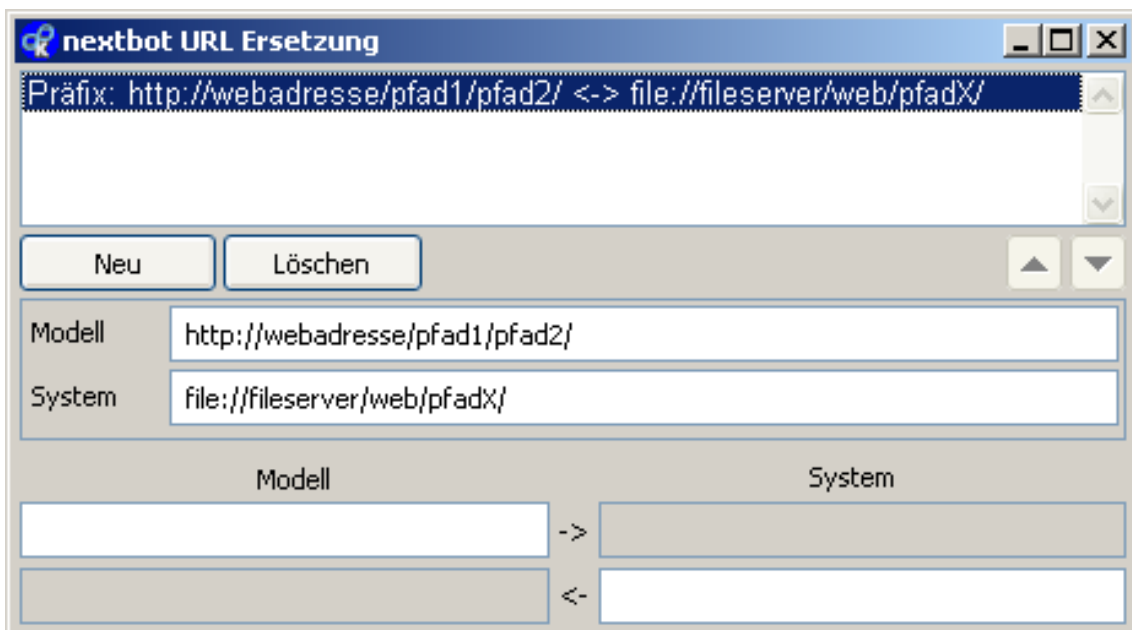
Zu diesem Zweck sind zwei unterschiedliche Ansätze umgesetzt:

- Präfix-Transformation
- Transformation durch reguläre Ausdrücke



Dabei ist zu beachten, dass in beiden Fällen eine Abbildung in beide Richtungen (Modell zu System und System zu Modell) formuliert werden muß, da eine Abbildung eines nextbot-Treffers, in dem ja der systemnahe Identifikator steht, auf den im Wissensnetz rückzuverfolgenden Modell-URL-Wert möglich sein muß.

Bei der Präfix-Transformation ist der letztendliche Dateiname für beide gleich, nur ein Pfad-Präfix muss für den Nextbot-Zugriff ersetzt werden:



Das korrekte Verhalten der Transformation läßt sich in den ganz unten angebrachten Testfeldern anhand von Beispieleingaben für beide Richtungen überprüfen.

Bei der Transformation über reguläre Ausdrücke können in der URL wie im Nextbot-Identifikator bestimmte Abschnitte definiert werden, die fix sind, und solche, die vom Ausgangspunkt in das gewünschte Resultat übertragen werden. Der Umgang mit regulären Ausdrücken sei an folgendem Beispiel erläutert:

Es soll von der URL `http://server:portnu/path1/path2/file.ext` transformiert werden auf die Nextbot-verständliche Angabe `\\netzlaufwerk\pfadX\path2\file.ext`. D.h. zu übernehmende Abschnitte sind `path2`, `file` und `ext`, die restlichen müssen nur jeweils erkannt werden.

Somit ist für die Identifikation der gewünschten Abschnitte folgender regulärer Ausdruck zu formulieren:



`http://server:portnum/path1/(.*/)(.*/)(.*)`

Das System liest diesen Ausdruck ein und erkennt diejenigen Abschnitte, die in die Ersetzung übernommen werden sollen daran, dass sie in runde Klammern eingeschlossen sind. Zu beachten ist auch, dass einige Zeichen für die Syntax von regulären Ausdrücken benutzt werden, solche Zeichen müssen durch das Voranstellen eines backslash "\" von der Betrachtung beim Einlesen ausgenommen werden (hier konkret der Doppelpunkt und der einfache Punkt). Eine genaue Beschreibung regulärer Ausdrücke findet sich bspw. hier:

`http://en.wikipedia.org/wiki/Regular_expression`

Nach der Eingabe des obigen Ausdrucks quittiert das System, ob ein gültiger Ausdruck eingegeben wurde zeigt eine grünen Zahl hinter der Eingabe an. Diese Zahl ist die Anzahl der erkannten Abschnitte plus 1 (plus 1 deshalb, weil der gesamte Ausdruck als Abschnitt 1 referenzierbar ist). Sie gibt auch den maximalen Indexwert vor, der dann in der Ersetzung verwendet werden kann.

Die Ersetzung muss danach in der Zeile darunter eingegeben werden. Für das verwendete Beispiel ist die Ersetzung des backslash von besonderer Bedeutung, dieser muss verdoppelt werden, wenn er als Zeichen angegeben werden soll. Um nun einen der zu übernehmenden Abschnitte zu referenzieren, verwendet man einen backslash gefolgt von der Indexzahl des Abschnitts plus 1. Daher muss der Ersetzungsstring wie folgt lauten:

`\\host\pathX\\2\\3.\\4`

Die Gültigkeit der Eingabe wird ebenfalls vom System überprüft und durch einen grünen Haken rückgemeldet.

nextbot URL Ersetzung

Präfix: `http://webadresse/pfad1/pfad2/ <-> file://fileserver/web/pfadX/`

Regex: `http://server:portnum/path1/(.*/)(.*/)(.*)` -> `\\\\host\\pathX\\2\\3.\\4`

Neu Löschen

Modell `http://server:portnum/path1/(.*/)(.*/)(.*)` 4

System `\\\\host\\pathX\\2\\3.\\4` ✓

System ✗

Modell -

Modell -> System

System <- Modell

Auch die Rückrichtung muss nun analog formuliert werden, für unser Beispiel lauten die zu tätigen Eingaben:

`\\\\host\\pathX\\(.*/)\\(.*/)(.*)`

`http://server:portnum/path1/2/3.4`

nextbot URL Ersetzung

Präfix: http://webadresse/pfad1/pfad2/ <-> file://fileserver/web/pfadX/
Regex: http\://server\:\portnum/path1/(.*)/(.*)\.(.*) -> \\host\pathX\\2\\3.4 / http

Neu Lösch

Modell 4
System ✓

System 4
Modell ✓

Modell System
 ->
 <-

2 Nextbot-Suchen

2.1 Konfiguration der Nextbot-Suche

Nachdem der nextbot-Index im Wissensnetz eingerichtet ist, kann man nextbot-Suchen konfigurieren und verwenden. Es gibt mehrere Typen von Nextbot-Suchen. Diese gliedern sich in Suchen nach Containern und Dokumenten. Jede Suche nimmt dabei verschiedene Eingabeobjekte auf oder hat unterschiedliche Suchlogik.

Es gibt unten beschrieben Suchen nach Dokumenten. Jeder Dokumenttreffer einer Suche kann - je nach Konfiguration - aus dem Netz stammen oder auch ein externer Treffer z.B. aus einen Verzeichnis sein.

1. nextbot Volltextsuche

Diese Suche liefert Dokumente, die einem angegebenen Suchstring entsprechen. Es werden alle Dokumente gefunden, die mindestens eines der Suchworte im Suchstring enthalten. Die Bewertung der Treffer richtet sich danach:

- wieviele der Suchworte getroffen wurden
- wie häufig ein Wort im Dokument getroffen wurde
- welche der Worte im Bezug zum Gesamtbestand relevanter sind.

2. nextbot Dokumenten Suche über einen Suchausdruck

Ein Suchausdruck kann eingegeben werden, der für jeden Term dessen Wichtigkeit bestimmt und Suchterme und Ergebnsgewichtung durch Verknüpfungs- und Gewichtungsooperatoren einstellt.

3. nextbot Dokumenten-Ähnlichkeitssuche

Findet ähnliche Dokumente zu einem Dokument.

4. nextbot Dokumenten-Ähnlichkeitssuche für einen Text



Hier wird ein Suchtext eingegeben und zu diesem Text passende Dokumente gesucht, die einen ähnlichen Inhalt aufweisen.

Alle Dokumentsuchen können Metaeigenschaften der Treffer liefern, die dann für eine Webanwendung abgefragt werden können.

Die Container suchen finden immer Objekte des Netzes:

1. nextbot Container-Ähnlichkeitssuche
Zu einer Menge von Dokumenten oder Containern werden passende Container gesucht. Dies ist die typische Suche für Themenvorschläge.
2. nextbot Container-Ähnlichkeitssuche mit Suchtext
Zu einem Suchtext werden Container gesucht, deren zugeordnete Dokumente ähnlich zum Suchtext sind.
3. nextbot Container Suche für einen Suchtext über ähnliche Dokumente
Diese Suche ähnelt der vorigen nur das zuerst Dokumente gesucht werden, die zu dem Suchtext passen und von diesen dann deren explizite Zuordnung zu Containern betrachtet wird.

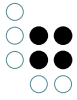
2.2 Durchführen der Nextbot-Suche

Die Durchführung der Nextbot-Suchen geschieht analog zu den anderen im Knowledge-Builder konfigurierbaren Suchen: Nach der Konfiguration einer Nextbot-Suche muß diese in die Menge der angebotenen Suchen übernommen werden, danach läßt sie sich über das Suchfeld und den Suchknopf starten.

3 KPath Funktionen

Dieses Kapitel beschreibt KPath Funktionen, die spezifische Erweiterungen für ein Netz mit nextbot Index sind. Alle Funktionen werden auf Topics angewendet.

Element	Beschreibung
<code>nbIsDocument(Topic (filterUnknownDocuments)</code>	Ergebnis: • true falls das Topic ein Dokument im Sinne von nextbot ist. Parameter: • filterUnknownDocuments ist optional und bestimmt, ob für das Dokument überprüft werden soll, ob nextbot dieses (noch) kennt (Boolean, Default: true).
<code>nbIsContainer()</code>	Ergebnis: • true, falls das Topic ein Container im Sinne von nextbot ist.



nbPri- ma- ry- Doc- u- ments	Ergebnis: - Die Menge der Trainingsdokumente für den gegebenen Container. Parameter: filterUnknownDocuments ist optional und bestimmt, ob für das Dokument überprüft werden soll, ob nextbot dieses (noch) kennt (Boolean, Default: true).
nbPro- mot- ed- By	Ergebnis: Die Menge der Container des Trainingsdokumentes.
nb- Size(lang)	Ergebnis: - Liefert die Größe des an ein Dokumenten-Topic angehängten Binärobjekts in der angegebenen Sprache lang in Byte (Integer). Ist ggfs. nil, wenn das Dokument nicht mehr existiert. Nur sinnvoll bei Topics, die für nextbot ein Dokument darstellen, ansonsten "nil". Parameter: lang ist optional, per Default wird die Sprache der Session genutzt.
nbLas- Mod- i- fied(lang)	Ergebnis: - Liefert einen Zeitstempel (Timestamp) das Datum der letzten Modifikation des ein Dokumenten-Topic angehängten Binärobjekts in der angegebenen Sprache lang. Nur sinnvoll bei Topics, die für nextbot ein Dokument darstellen, ansonsten "nil". Auch "nil", wenn das Dokument nicht (mehr) im nextbot Index existiert. Parameter: lang ist optional, per Default wird die Sprache der Session genutzt.
nb- Snip- pet(lang)	Ergebnis: - Liefert einen String, der den Anfang des an ein Dokumenten-Topic angehängten Binärobjekts in der angegeben Sprache lang. Nur sinnvoll bei Topics, die für nextbot ein Dokument darstellen, ansonsten "nil". Auch "nil", wenn das Dokument nicht (mehr) im nextbot Index existiert. Parameter: lang ist optional, per Default wird die Sprache der Session genutzt.



nbFeatures(count, showWeights, lang)	Ergebnis: • Liefert einen String mit einer Liste von nextbot Features für das an ein Dokumenten-Topic angehängten Binärobjekts in der angegeben Sprache lang. Nur sinnvoll bei Topics, die für nextbot ein Dokument darstellen, ansonsten "nil". Auch "nil", wenn das Dokument nicht (mehr) im nextbot Index existiert. Parameter: • count ist optional und bestimmt die maximale Anzahl an Features (Integer, Default: 12). • showWeights ist optional und bestimmt, ob die Gewichtung der einzelnen Features angezeigt werden soll (Boolean, Default: false). • lang ist optional, per Default wird die Sprache der Session genutzt.
nbRepositoryType(lang)	Ergebnis: • Art des Repositories (String) "internal" oder "external", wobei interne vom VFS stammen und externe durch Web-Crawling in den Index gelangt sind. Nur sinnvoll bei Topics, die für nextbot ein Dokument darstellen, ansonsten "nil". Auch "nil", wenn das Dokument nicht (mehr) im nextbot Index existiert. Parameter: • lang ist optional, per Default wird die Sprache der Session genutzt.
nbRepository(lang)	Ergebnis: • Liefert den Namen des Repositories. Bei internen Dokumenten ist dies der Kurzname, wie er in der nextbot Konfiguration angegeben wurde. Bei externen Dokumenten ist dies der Hostname der URL. Nur sinnvoll bei Topics, die für nextbot ein Dokument darstellen, ansonsten "nil". Auch "nil", wenn das Dokument nicht (mehr) im nextbot Index existiert. Parameter: • lang ist optional, per Default wird die Sprache der Session genutzt.
nb-System-Locator(lang)	Ergebnis: • Liefert den Locator, der intern im System genutzt wird, z.B. zum Herunterladen des Dokuments. Nur sinnvoll bei Topics, die für nextbot ein Dokument darstellen, ansonsten "nil". Auch "nil", wenn das Dokument nicht (mehr) im nextbot Index existiert. Parameter: • lang ist optional, per Default wird die Sprache der Session genutzt.



nbFile Name(lang)	Ergebnis: • Diese KPath-Funktion kann an einem Individuum aufgerufen werden, welches ein von Nextbot indexiertes Dokument repräsentiert, um den Dateinamen dieses Dokuments abzufragen. Parameter: • lang ist optional, per Default wird die Sprache der Session genutzt.
nbFile Extension(lang)	Ergebnis: • Diese KPath-Funktion kann an einem Individuum aufgerufen werden, welches ein von Nextbot indexiertes Dokument repräsentiert, um die Dateierweiterung des Dokuments abzufragen. Parameter: • lang ist optional, per Default wird die Sprache der Session genutzt.
nbO- rig- i- nal- File()	Ergebnis: • Diese KPath-Funktion kann an einem Individuum aufgerufen werden, welches ein von Nextbot indexiertes Dokument repräsentiert, um den Ort herauszufinden, an der das indexierte Dokument liegt, beispielsweise ein Ort auf einem File-share.
nbO- rig- i- nalDirec- to- ry()	Ergebnis: • Diese KPath-Funktion kann an einem Individuum aufgerufen werden, welches ein von Nextbot indexiertes Dokument repräsentiert, um das Verzeichnis herauszufinden, in dem das indexierte Dokument gespeichert ist.
nbCar Be- Pro- mot- ed(topicLocator)	Ergebnis: • true / false. Diese KPath-Funktion kann an einem Hit aufgerufen werden und ermittelt, ob dieser an das gegebene Topic promoted werden kann. Parameter: • topicLocator ist ein String, der das Topic spezifiziert.
nbCar De- mot- ed(topicLocator)	Ergebnis: • true / false. Diese KPath-Funktion kann an einem Hit aufgerufen werden und ermittelt, ob dieser für das gegebene Topic demoted werden kann. Parameter: • topicLocator ist ein String, der das Topic spezifiziert.



4 KEM

4.1 KEM Funktionen

Promote und Demote von Topics

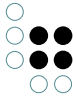
Name	KEMTopicEditor:: promoteDocumentTopic(documentEditorRef)
Beschreibung	Liefert einen KEMSelectableListEditor mit möglichen Relationen. Durch select() der entsprechenden Relation wird das Promote ausgelöst. Rückgabewert ist die neu gezogene Relation.

Name	KEMTopicEditor:: promoteDocumentLocator(locatorString)
Beschreibung	Liefert einen KEMSelectableTreeNodeEditor, der auf der ersten Ebene die möglichen Promote-Relationen liefert und auf der zweiten Ebene die möglichen Dokument-Begriffe. Durch select() des gewünschten Begriffs, wird ein Dokument erstellt und die entsprechende Relation gezogen. Rückgabewert ist das neu erzeugte Dokumentindividuum, welches den übergebenen locatorString als URL und Namen hat.

Name	KEMTopicEditor:: demoteDocumentTopic(documentEditorRef)
Beschreibung	Liefert einen KEMSelectableListEditor mit Relationen. Durch select() der entsprechenden Relation wird das Demote (i.e. Löschen der Relation) ausgelöst. Rückgabewert ist nil.

Information zur Konfiguration von nextbot

Name	KEMSession:: getContainerSuperConcepts()
Beschreibung	Liefert einen KEMListEditor mit den Begriffen, deren Unterbegriffe in NextBot als Container behandelt werden.



Name	KEMSession:: getContainerInstanceConcepts()
Beschreibung	Liefert einen KEMListEditor mit den Begriffen, deren Individuen in NextBot als Container behandelt werden.

Name	KEMSession:: getDocumentInstanceConcepts()
Beschreibung	Liefert einen KEMListEditor mit den Begriffen, deren Individuen in NextBot als Dokument behandelt werden.

Name	KEMSession:: isExternalIndexAvailable()
Beschreibung	True wenn ein NextBot-Index vorhanden und aktiv ist.

Name	KEMUser:: getRepositoryNames()
Beschreibung	Liefert eine Liste der Namen der innerhalb von nextbot konfigurierten externen Repositories. Das Pseudo-Repository „VOLUME“ ist nicht enthalten.

Download von Dokumenten per Stream

Name	KEMUser:: getExternalDownloader(locator)
Beschreibung	Liefert einen Downloader mit dem man aus diesem Volume für diesen User das Objekt unter dem übergebenen Locator geliefert bekommt.



Name	KEMInstanceEditor:: getExternalDownloader()
Beschreibung	Für dieses Dokumentenindividuum einen Downloader bekommen. Standardmäßig wird das Dokument in der Sprache der Session gesucht, falls in dieser Sprache kein Dokument vorliegt werden auch andere Sprachen überprüft.

Name	KEMInstanceEditor:: getExternalDownloaderForLanguage(language)
Beschreibung	Für dieses Dokumentenindividuum einen Downloader bekommen. Es wird das Dokument in der angegebenen Sprache gesucht oder kein Fehler verursacht, falls es in dieser Sprache nicht vorliegt.

4.2 Eigenschaften von Suchen

Die im folgenden aufgezählten Filter sind nur bei Suchen sinnvoll, deren Ergebnisobjekte Dokumente darstellen.

Sortierung durch Feedback beeinflussen

Um die Sortierung der Ergebnisse einer NextBot-Suche über KEM zu beeinflussen, können folgende Parameter gesetzt werden:

- positiveFeedbackTopics
- negativeFeedbackTopics
- positiveFeedbackLocators
- negativeFeedbackLocators

Diese beiden „*feedbackTopics“ Methoden nehmen als Parameter jeweils eine Liste von Topics, die als positive bzw. negative Beispiele verwendet werden. Für externe Treffer (ohne Topic Entsprechung im Netz) können mit den nächsten zwei „*feedbackLocators“ Methoden deren Locator verwendet werden.

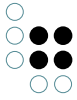
Filtern nach Repository

Um das Suchergebnis vorab durch nextbot auf bestimmte Quellen (Repositories) einzuschränken, kann bei Suchen, die Dokumente finden, der Parameter

- repositoryFilter

gesetzt werden. Wenn kein Filter oder eine leere Liste von Strings übergeben wird, so findet kein Filtern statt. Ungültige Repository Namen werden ausgefiltert.

Filtern nach Dateityp

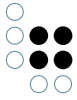


Bei Suchanfragen kann eine Menge von Dateitypen (Dateinamensendungen) übergeben werden, aus der die Treffer stammen sollen. Treffer von Dateien mit anderen Endungen werden dann nicht zurückgeliefert.

- filetypeFilter

4.3 Eigenschaften von Suchergebnissen

Name	Typ	Beschreibung
quality	Float	Die Qualität des Treffer im Bereich [0..1]. Die Qualität ist relativ zur Qualität anderer Treffer in diesem Suchergebnis zu verstehen. Sie ist nicht direkt vergleichbar zu Qualitätswerten, die man aus weiteren Suchen erhält.
locator	String	Ein eindeutiger Bezeichner des Objekts, das hier getroffen wurde. Typischerweise ist dies die URL unter der das getroffene Objekt angesprochen werden kann. Je nach Konfiguration kann dies aber eine andere Bedeutung haben.
label	String	Kurzbezeichner des Objekts, typischerweise der Dateiname, den man zur Anzeige verwenden möchte.
repository-Type	String	Ang der Quelle, aus der der Treffer stammt. Mögliche Werte sind „internal“ und „external“, wobei „internal“ alle Treffer bezeichnet, die aus eigenen Datenbeständen stammen und „external“ Treffer aus Fremdbeständen (z.B. er-crawl-te Webseiten und Dokumente).
repository	String	Name der Quelle, aus der das Dokument stammt. Für „internal“ Quellen ist das entweder „VOLUME“ für Dokumente, die nur im Wissensnetz existieren oder der Name der Quelle, wie er in der Konfiguration angegeben wurde. Bei „external“ Quellen ist es der Name des Rechners von dem der Treffer stammt.
language (optional)	String	Die Sprache des Treffers, sofern diese ermittelt werden konnte.
size (optional)	Integer	Die Größe der Quelldatei des Treffers in Byte, sofern diese ermittelbar ist.
lastmodified (optional)	Timestamp	Das Datum der letzten Modifikation des Treffers.
snippet (optional)	String	Beispielhafter Ausschnitt aus der getroffenen Datei, in dem mindestens eines der Suchworte vorkommt.



start-Snip-pet (optional)	String Text vom Anfang der Datei.
file-Name	String Dateiname des Dokuments; Bei Blob-Attributen die über Nextbot indexiert werden: Dateiname, der im Wissensnetz für das Blob-Attribut hinterlegt ist.
fileEx-tension	String Dateierweiterung (Dateityp) des Dokuments; Bei Blob-Attributen: Dateierweiterung des Dateinamens, der im Wissensnetz für das Blob-Attribut hinterlegt ist.